EMBEDDED SYSTEM

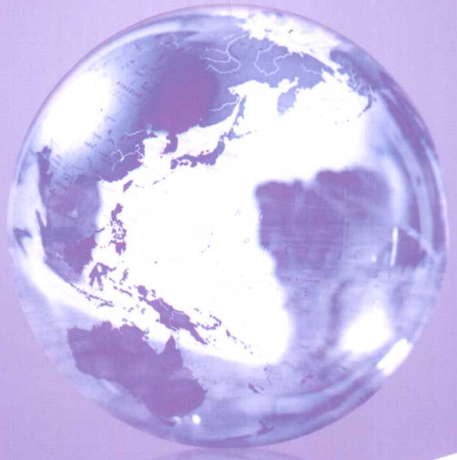
嵌入式技术与应用丛书

计算机图形

显示、加速及实现技术

——基于VxWorks的嵌入式图形系统开发实例

赵刚 张翀 江勇 著



完整的嵌入式图形系统软硬件设计方案

从驱动层面剖析图形处理器GPU硬件加速原理

VxWorks下的OpenGL驱动程序开发方法及驱动实例

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

计算机图形

嵌入式技术与应用丛书

计算机图形显示、加速及实现技术

——基于VxWorks的嵌入式图形系统开发实例

赵 刚　张 翀　江 勇 著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

本书实现了一个完整的嵌入式图形系统软硬件设计方案，讲解了从驱动层面剖析图形处理器GPU硬件加速原理，详细介绍了VxWorks下的OpenGL驱动程序开发方法及驱动实例。

本书共分为9章。第1章介绍了计算机图形的基本概念以及计算机图形系统的总体结构和工作流程。第2章讲述了计算机图形显示原理。第3章着重讲述了图形处理芯片（GPU）的基本架构及加速原理。第4章讲述了通用图形系统的软硬件实现原理及基本开发流程。第5章结合图形理论基础和嵌入式系统的特点，指出了在嵌入式图形系统开发中所需关注的若干要点。第6章介绍了嵌入式图形系统硬件设计原理及实现方法。第7章对嵌入式图形系统开发中所要用到的软件开发环境、开发工具及其具体使用方法进行了详尽的介绍。第8章详细讲述了嵌入式图形系统驱动程序的开发。第9章介绍了图形系统OpenGL驱动开发的基本方法。

本书是在作者多科研工作基础上完成的，其中包含了大量最新的科研成果与应用经验，力求创新性、先进性和实用性。

本书主要提供给研究所和企业的IT产品研发人员作为技术参考，亦可作为电子信息类研究生的相关课程教材。

图书在版编目（CIP）数据

计算机图形显示、加速及实现技术——基于VxWorks的嵌入式图形系统开发实例/赵刚等著. —北京：电子工业出版社，2009.6
（嵌入式技术与应用丛书）
ISBN 978-7-121-08908-4

Ⅰ. 计… Ⅱ. 赵… Ⅲ. ①计算机图形学 ②实时操作系统，VxWorks Ⅳ. TP391.41 TP316.2

中国版本图书馆CIP数据核字（2009）第081057号

责任编辑：高买花 田宏峰 特约编辑：刘 涛
印 刷：北京市智力达印刷有限公司
装 订：北京中新伟业印刷有限公司
出版发行：电子工业出版社
北京市海淀区万寿路173信箱 邮编 100036
开 本：787×1 092 1/16 印张：15.5 字数：340千字
印 次：2009年6月第1次印刷
印 数：4 000册 定价：36.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

前 言

当你利用计算机玩游戏或欣赏动画片时，屏幕上所呈现出的五彩缤纷的画面、逼真的三维图形会时常让你为之折服，叹为观止。当人们赞叹计算机所具有的这一切神奇功能时，许多人并不知道，计算机内有一颗图形处理器（也叫做图形处理器或图形处理单元）（GPU，Graphics Processing Unit）芯片正默默地高速运行着。也正因为 GPU 的诞生，才使得现代计算机屏幕变得更加生动逼真，为人们带来了巨大的视觉冲击与享受，进一步推动了计算机技术的发展与普及。

目前，GPU 在通用计算机中已得到了广泛的应用。几乎每一台计算机上都配置了一块显卡，而每一块显卡的核心芯片就是 GPU。作为 3D 图形硬件基础，图形处理器（GPU）在整个计算机图形系统中发挥着重要的作用，其发展速度大大超过了计算机系统中其他硬件单元的发展速度。从 1999 年 NVIDIA 公司推出第一款具有 3D 图形加速功能的 Geforce256 图形芯片，到 2005 年推出的具有强大的图形管道可编程能力的 Geforce7 系列的图形芯片，图形芯片经历了五代的革新，而在此期间，英特尔公司的中央处理器芯片的发展仅仅从 Pentium Ⅲ发展到 Pentium Ⅳ。图形处理器的集成度也已大大超越了中央处理器的集成度，其发展更新速度超出了摩尔定律的限制。在 GPU 强大图形处理能力的支持下，大型三维电脑游戏、网络游戏以及 Play Station、Xbox 等新型游戏平台正试图将人们从传统的电视、电影等娱乐方式中剥离出来。所有这些都预示着图形技术将成为 21 世纪最具活力的研究领域。除此之外，GPU 的发展极大地提高了计算机图形处理的速度和图形质量，并促进了虚拟现实、计算机动画和自然景物仿真等领域的快速发展。图形处理器绘制流水线高速、并行的特性以及灵活的可编程功能，为数字图像处理和通用并行计算提供了良好的运行平台。作者凭借多年来的相关科研成果与经验，撰写完成目前国内首部 GPU 开发应用的技术专著，期望推动 GPU 在广大 IT 产品，特别是在嵌入式产品中得到运用，为我国的经济建设服务。

全书共分为 9 章。前 4 章构成本书上篇，着重介绍计算机图形显示及其加速原理，以及通用图形系统中的软硬件构成和开发流程，使初学者掌握相关概念，为后一阶段的 GPU 实际开发打下良好的基础。后 5 章构成本书下篇，通过一个实际的 GPU 应用开发实例，对嵌入式图形系统的构成、GPU 内部架构、开发工具、驱动程序编写进行了详细地描述，其中包含了大量最新的科研成果与应用经验。读者通过下篇的学习，应能建立起对嵌入式图形系统开发流程的全新认识。在实际的开发过程中，尽管读者所选用的 GPU 具体型号、操作系统、开发软件可能与本书实例有所不同，但开发方法、开发流程却是完全相同的，以达到能举一反三的目的。

第 1 章介绍了计算机图形的基本概念以及计算机图形系统的总体结构和工作流程。第 2 章讲述了计算机图形显示原理，包括基本图元的生成、图元在屏幕上的变换以及真实感图形的显示，通过实例展示了计算机图形绘制的全过程。第 3 章着重讲述了图形处理器

（GPU）的基本架构及加速原理。第 4 章讲述了通用图形系统的软硬件实现原理及基本开发流程。第 5 章结合上篇的图形理论基础和嵌入式系统的特点，指出了在嵌入式图形系统开发中所需关注的若干要点。第 6 章介绍了嵌入式图形系统硬件设计原理及实现方法。第 7 章对嵌入式图形系统开发中所要用到的软件开发环境、开发工具及其具体使用方法进行了详尽地介绍。第 8 章详细讲述了嵌入式图形系统驱动程序的开发。第 9 章介绍了图形系统 OpenGL 驱动开发的基本方法。

本书主要提供给研究院所和企业的 IT 产品研发人员作为技术参考，亦可作为电子信息类研究生的相关课程教材。

本书编写组由赵刚、张翀、江勇三位教师组成，编写工作由参加编写的教师集体讨论，分工编写、交叉修改，历时两年完成。赵刚担任主编，负责大纲拟定、组织编写与统稿工作。

在编写过程中，得到四川大学电子信息学院的杨鹏、黄赞、陈雷等研究生的大力支持和帮助。借本书出版之际，向他们表示衷心的感谢！

虽然本书是在作者多年科研工作基础上完成的，但由于作者水平有限，疏漏之处在所难免，敬请读者批评指正（E-mail：zgscu@163.com）。

作 者

2009 年 6 月于四川大学

目　　录

上篇 计算机图形显示及其加速原理

上　篇

计算机图形显示及其加速原理

在计算机中，一个图形到底是怎样显示到屏幕上的？它在显示屏幕上又是如何定位的？计算机是怎样实现对一个图形的移动、旋转、缩放操作的？二维图形和三维图形在实现原理上又有什么不同？图形的颜色、纹理、光照效果是怎样实现的？计算机图形系统的各个部分是如何协调一致工作的？这些问题常常会在图形开发初学者的脑海中萦绕。本篇将为你逐层揭开上述疑团，从而在你的脑海中建立起一个图形系统的基本框架。

上篇由第 1 章到第 4 章组成。第 1 章从系统的角度描述了计算机图形的基本知识和计算机图形系统的工作流程。第 2 章从计算机图形学的角度讲述了图形的显示原理。第 3 章从图形处理器（GPU）的基本架构和工作流程的角度，描述了图形加速的基本原理。第 4 章通过对通用图形系统架构和开发方法的描述，为读者建立起一个完整的图形系统概念。

第 1 章　计算机图形概述

图形作为人类传递信息的主要方式之一，已被广泛地应用于各类信息系统中。在人机交互系统中，图形输出为人们提供最直观的交互方式。图形是如何通过计算机产生的？从最初的几何数据参数到最终的图形显示将经历怎样的一个过程？计算机图形有哪些特点？以上问题都将在本章中找到答案。

本章将首先介绍计算机图形与计算机图像的区别，然后介绍作为计算机图像载体的计算机图形系统的结构、功能及发展，最后详细描述从图形几何参数到生成显示在屏幕上的图形的全过程。

1.1　计算机图形与计算机图像

1.1.1　计算机图形与计算机图像的区别

计算机图形与计算机图像是两个容易混淆的概念。计算机图形是由基本几何图元（包括点、线、面、体等）组成的图元集合，用以描述物体表面特征。构成计算机图形的基本几何图元是由计算机在数学模型基础上通过一定的算法生成的。计算机图像是由扫描仪、摄像机等输入设备捕捉实际画面产生的数字图像，是由像素点阵构成的位图。

从数据流的角度来看，计算机图形是用一组指令集合（实质上是几何图元的数学方程）来描述图形的内容，如描述构成该图的各种图元端点位置、形状、色彩等，因此描述对象可任意缩放而不会失真；计算机图像则是用数字描述每个像素点的色彩、亮度、饱和度等属性，描述其信息的文件占用较大的存储空间，所描述对象在缩放过程中会损失细节或产生锯齿。计算机图形与计算机图像的区别如表 1.1 所示。

表 1.1　计算机图形与计算机图像的区别

比较项目	计算机图形	计算机图像
基本描述单元	几何图元	像素点
描述方式	数学方程	像素点色彩、亮度、饱和度
占用存储空间	较小	较大
处理效果	缩放不失真	缩放失真，图像模糊或出现马赛克

续表

适用范围	能准确描述 3D 视图，但较难描述自然景物	部分 3D 信息已经丢失，但能真实自然地描述自然景物
常用文件格式	.SWF,WMF,DXF,CGM,SVG	.BMP,JPG,PSD,GIF,PDF,TIF
常用处理软件	AutoCAD,FreeHand,Illustrator,CorelDraw	Photoshop,PhotoImapct,Paint Shop Pro,Fractal Design'Painter

由表 1.1 可见，计算机图形与计算机图像之间存在着较大的区别。因此，对计算机图形的处理和对计算机图像的处理是两个不同的过程，对于一个计算机图形系统，其输入为几何图元数据，输出为具体的图形，处理流程如图 1.1 所示。

而对于一个计算机图像系统，输入为原始的图像，根据需要对其进行处理（去噪、增强、复原、重建、编码、压缩、重建或传输等），最后输出处理后的图像，其处理流程如图 1.2 所示。

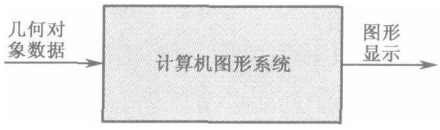


图 1.1　计算机图形系统处理流程

图 1.2　计算机图像系统处理流程

1.1.2　点阵图形和矢量图形

图形可以分为矢量图形和点阵图形两大类。最先出现的计算机图形被称为点阵图形，也叫做位图图形。这个名称直接反映出了计算机图形的物理构成——显示器的屏幕由一些可以发光的像素点组成，这些像素点构成了一个矩形的阵列。通过编程控制这些像素哪些被点亮，就在屏幕上形成了最初的图形。

（1）点阵图形

对于一个分辨率为 1 024×768 显示器屏幕而言，像素在其上的排列方式如图 1.3 所示。

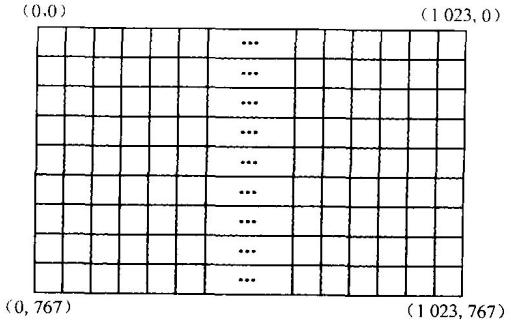


图 1.3　点阵图形像素排列方式

通常，图形开发人员使用分辨率来描述全屏幕显示的点阵图形的大小。常见的分辨率有 640×480,800×600,1 024×768 等。如果分辨率为 1 024×768，那么从屏幕左上角开始，在水平方向上一行共有 1 024 个像素点，在垂直方向上一列共有 768 个像素点。为了准确

地描述每一个像素点，可以用一个类似于二维坐标系（xy 坐标系）的标识方法来确定每个像素点的位置。第 1 个像素点为（0，0），第 2 个像素点为（1，0），第 1 024 个像素点为（1 023，0），第 1 025 个像素点为（0，1）…，而位于屏幕右下角的最后一个像素点为（1 023，767）。由于一个像素的位置是由水平（x）方向和垂直（y）方向共同决定的，因此从数学描述的角度来看，点阵图形也可以称为二维图形。当然一个实际的点阵图形可以小于全屏幕显示的点阵图形，也可以大于全屏幕显示的点阵图形。

如上所述，像素是构成点阵图形的最小元素，那么一个像素究竟是什么形状呢？如果把一个像素简化为一个点。从数学的角度分析，一个点只有位置，没有大小和形状之说。既然是要从图形上表示出一个点，通常情况下像素在屏幕上的形状可以视为图 1.3 中的一系列方格的组合，只是因为这些方格足够小，肉眼无法识别其大小和形状而已。

明白了像素的形状，还需要搞清楚像素的颜色是怎样确定的。首先来看最简单的黑白点阵图形。每个像素只需要用一个 bit 即可以表示出黑白图形来。说得更形象些，每个像素都是一个灯泡，开关闭合时灯泡被点亮成白色，开关断开时为黑色。假如用“1”来表示开关闭合，用“0”来表示开关断开，那么通过简单的 01 序列即可形成黑白点阵图形。

以此类推，如果用一个字节来表示一个像素的颜色，那么一个像素就可以有 $2^8 = 256$ 种颜色。这就是通常所说的 256 色显示器。随着计算机技术的发展，目前已经开始使用 4 个字节来表示一个像素颜色了。这 4 个字节分别表示 RGB 三基色以及 Alpha 颜色通道，能表示的颜色数目已超过 1 600 万。事实上，人的肉眼所能分辨的颜色远远小于这个数字。也就是说，现有的颜色种类已足以满足人们在视觉方面的需求。

此外，点阵图形中所用到的颜色种类数也是决定一个点阵图形文件大小的重要因素。点阵图形颜色的种类越多，表达每个像素点颜色所占用的存储器的位数也就越多，在图形点阵数量相同的前提下，点阵图形文件也就越大。例如，图形大小相同的两个点阵图形，采用 256 色的点阵图形，其文件大小将是采用黑白两色点阵图形的 8 倍；如果采用 24 位真彩色的点阵图形，则文件大小将是采用黑白两色点阵图形的 24 倍。

（2）矢量图形

显然，用点阵图形来表示某个特定的图形时，图形的结构显得非常简单：图形是由许许多多具有不同颜色的点构成的。绘出某个图形所需要的主要工作也就是在指定的图形显示介质上指定各个像素点的颜色。这种绘制图形的方式在某些场合下对图形开发人员来说就会显得很不方便。比如绘制一条直线段，斜率不同，要填充的像素点就不同；绘制一个圆，圆心位置和半径的大小不同，所要填充的像素区域也会有所不同。于是图形开发人员开始设计一种新的绘制图形的方式。仍以画一条直线段为例，由数学几何公理可知，两点决定一条直线，如果只需编程人员指定出一条直线段的起点和终点，剩下的所有点都可以通过数学模型由计算机自动给出，这种绘图方式将极大减轻图形开发人员在编写程序上的复杂度。再比如绘制一个圆，图形开发人员只需要提供圆心坐标和半径长度参数，计算机就会根据数学模型自动计算出需要填充的像素点并赋予指定的颜色，这种绘制图形的方式必将大大提高开发人员的编程效率。基于上述思路，一种不同于点阵图形的图形描述方式

产生了——用数学公式来描述图形：给出一些绘制图形所必需的几何参数，选择适当的数学模型，然后依此模型生成绘制几何图形的计算机命令，最终绘制出图形来。这种新方式下生成的图形被称为矢量图形（向量图形）。

如果说点阵图形与屏幕像素的是一一对应的话，矢量图形则与由各种数学公式所描述的基本图元（如线、三角形、四边形、圆等）是相对应的。矢量图形不仅可以存储二维图形（平面图形），还可以存储三维图形（立体图形）。那么如何确定一个矢量图形的颜色呢？还是先来看看点阵图形的颜色是如何确定的。如前所述，点阵图形由许许多多的像素构成，每个像素都可以有自己的颜色。与此类似，向量图形是由各种基本图形（基本数学公式）构成的，因此，各个基本图形都可以有各自独立的颜色。也就是说，构成一个矢量图形的基本图元（如直线段、三角形、正方形等）里的所有点都对应一个颜色。由于矢量图形的颜色是与构成矢量图形的基本图元相对应（不是屏幕上的像素点），因此，无论采用什么样的颜色模式都不会改变矢量图形的文件大小。

矢量图形最大的优点就是其本身是由精确的数据所给出的，因此可充分利用各种输出图形设备的分辨率来尽可能精确地输出图形，图形尺寸可以任意改变而不损失图像质量。相反，输出图形设备在输出点阵图形时，其光滑度就会受到输出图形设备分辨率和点阵图形本身分辨率的双重限制。由于点阵图形详细地规定了组成其自身的像素点的个数，像素点数随着不同的输出设备分辨率而可能改变，因此点阵图形不能任意放大或者缩小。当点阵图形放大到足够使人的肉眼分辨出单独一个像素点时，甚至有可能不能得到原始图形的外貌轮廓，一个最为常见的例子就是在电视节目中为了隐藏某些画面的细节而采用的马赛克技术。

（3）点阵图形与矢量图形的相互转换

尽管这两类图形各有自己的文件格式，但两者之间也是可以互相转换的。总地来说，不同点阵图形文件格式之间的转换要比点阵图形文件格式与矢量图形文件格式间的转换容易。点阵图形文件是一种按指定方法组织起来的一列像素点数据，结构简单。对这种文件进行转换时，只要从原点阵图形中读出像素点的数据，换成新的格式再写入文件就可以了。对于点阵图形而言，一个新的格式仅仅意味着像素点数据组织方式不同而已。

将点阵图形转换成矢量图形要复杂一些，可以采用两种方法。一种方法是把点阵图形转换成点阵物体，即把点阵图形作为一个基本图形嵌入到矢量图形中。这种方法的原理类似于在通常情况下将一幅画粘贴在不同的背景中。另一种方法是将点阵图形中用某种特征划分的各组像素点转换成与基本几何图形类似的矢量图形。可以选用 CorelTrace 这类软件跟踪点阵图形，然后分别对具有相同颜色的像素点建立起向量轮廓线，这样就可实现点阵图形到矢量图形的转换。

最为常见的也是用得最多的是将矢量图形转换成点阵图形。人们在显示器上所看到的矢量图形实际上是已经被转换成点阵图形的矢量图形，点阵打印机输出的也是已经转换成点阵图形的矢量图形。将矢量图形转换成点阵图形可按以下方式进行：首先通过软件解释文件中的向量命令，确定矢量图形的整体构成（包括物体与物体间的空间位置关系），然

后再建立该图的点阵格式图形。矢量图形在转换成点阵图形时需要指定点阵图形的分辨率，分辨率越高，意味着转换成的点阵图形的质量就越好，转换时间也会越长。

1.2 计算机图形系统的结构与功能

1.2.1 计算机图形系统的结构

计算机图形系统是一个可以将数据输入转换成图形输出的有机整体。作为计算机图形的载体以及其处理平台，计算机图形系统对于计算机图形的生成和处理起着至关重要的作用。第一台图形设备是1950年美国麻省理工学院（MIT）研制的“Whirlwind 1”(“旋风1号”)。实际上它是计算机的一个配件——图形显示器，只能显示简单的图形，类似一台示波器。之后经过半个世纪的长足进步，才有了如今日臻完善的计算机图形系统。

一个完整的计算机图形系统由图形硬件系统和图形软件系统两部分组成，传统的图形硬件系统包括图形处理器（GPU）、图形存储设备、图形输入设备和图形输出设备四个部分，它们之间的相互关系如图1.4所示。

在图形硬件系统中，GPU处于整个硬件系统的核心，根据图形渲染（通过计算机命令生成图形的过程称为渲染）命令处理各种二维或者三维的图形，并将经过处理后的图形送入帧缓冲区（存储图形数据的显存或内存），最后输出到显示器。常见的图形输入设备有键盘、鼠标、光笔、数字化仪、图形扫描仪等，常用图形输出设备有图形显示器、绘图仪、打印机等。

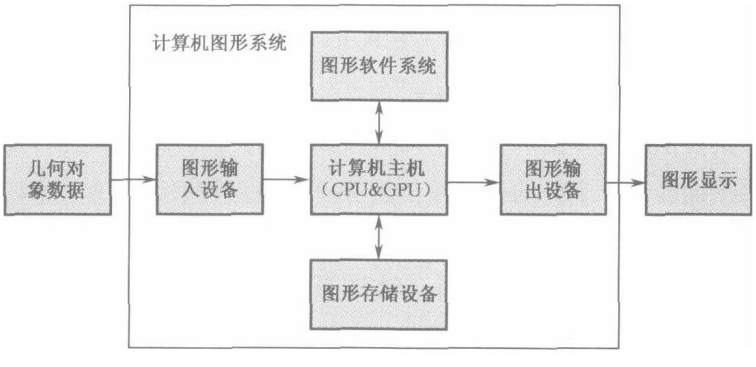


图1.4 计算机图形系统的结构示意图

1.2.2 计算机图形系统的基本功能

作为一个完整的系统，图形系统应至少能够完成计算、存储、交互、输入和输出五大功能。

计算功能包括形体设计、分析方法的程序库和有关描述形体的图形数据库。在图形程

序中应有坐标的几何变换、曲线和曲面的生成、图形的交点和接点的计算以及检验功能。存储功能指在计算机内存或外存上存放图形数据，尤其是存放图形数据之间的相互关系，并且可以根据需要实现信息的检索、增加、删除和修改等功能。交互功能是指通过硬件设备完成人机之间的通信。用户可以通过显示屏幕观察图形结果，用光笔或鼠标、键盘对不满意的图形进行修改，对图形中的错误进行提示、跟踪与检查。输入功能是指可以将图形的形状尺寸、位置、属性等参数通过输入设备输入至计算机内。输出功能是指将处理结果予以显示、打印和保存。由于输出结果在精度、形式及时间上有一定要求，可根据不同的输出设备予以满足。

1.3　计算机图形系统的发展

计算机图形系统的发展与计算机图形学的发展紧密相连。1963 年，美国麻省理工学院林肯实验室的 Ivan. E. Sutherland 发表了一篇题为“Sketchpad：一个人机通信的图形系统”的博士论文，论文中首次使用了“Computer Graphics”这个术语。这一名为 Sketchpad 的系统采用的是 TX-2 型计算机及阴极射线管式图形显示器，可用光笔在图形显示器上实现选择、定位等交互功能。计算机还可以跟踪光笔，从原来所在的点到所指定的点画出直线，或者在给定圆心和半径参数后画出圆来。这一系统还引入了分层存储符号和图素的数据结构，一幅完整的图可以通过分层调用若干子图来产生，这些基本概念和技术直至今日还在使用。该篇论文充分论证了对交互式计算机图形学进行研究的可行性以及重要性，从而确立了计算机图形学的学科地位。

1.3.1　图形系统硬件发展历程

作为图形系统硬件的核心，图形处理器（GPU）的主要任务是对输入的几何对象数据或是视频信息进行处理，如几何变换、裁剪、光照、光栅操作、纹理映射等。GPU 的出现，其初衷是为了减轻 CPU 日益繁重的图形处理任务，使得 CPU 更能专注于通用控制。GPU 的发展历程如图 1.5 所示，GPU 每一次突破性发展，实质上都是将一部分原本属于 CPU 的处理功能从 CPU 剥离出去，在自己身上重新实现并且对处理性能予以进一步的优化提高。

从早期功能单一的图形适配器到现在具有强大图形处理能力的 GPU，显卡的发展大致经历了以下六个阶段。

（1）显卡的雏形——从 MDA 到 CGA

早期的显卡称为显示适配器，最早的显示类型被称为 MDA（Monochrome Display Adapter），只能区别出黑白两色，一般集成 16 KB 显存，能够完成简单的字符及几何图形的显示。